DB Project Team 1

INF 280a – Fall 2015

SQL Queries

Implementation:

Query 1. Calculate the revenue of each dessert and sort them in descending order:

```
SELECT Dessert_D_ID,
       SUM(Price * Quantity) AS Revenue
FROM dessert_soldto_customer
GROUP BY Dessert_D_ID
ORDER BY 2 DESC
```

Query 2. Group the dependents by the employee they depend on and sort them by dependent name:

```
SELECT *
FROM dependent
WHERE Relationship IN ('Son',
                       'Daughter')
GROUP BY Employee_E_SSN
ORDER BY D_name
```

Query 3. Calculate the total amount of expenses incurred by each department:

```
SELECT department.Name,
       SUM(expense.Amount) AS Total_Expense
FROM ((department
       INNER JOIN department_incurs_expense)
      INNER JOIN expense)
WHERE ((department.D_number =
department_incurs_expense.Department_D_number)
       AND (department_incurs_expense.Expense_Ex_ID =
expense.Ex_ID))
GROUP BY department.Name
ORDER BY 1
```

Query 4. Calculate the total cost of each ingredient received from all suppliers in 2015:

```
SELECT Ingredient_I_ID,
       SUM(Price * Quantity) AS Total_Cost
FROM supplier_provides_ingredient
WHERE (Order_date BETWEEN 20150101 AND 20151231)
GROUP BY Ingredient_I_ID
ORDER BY Total_Cost DESC
```

Query 5. Calculate the total electricity consumption of each facility:

```
SELECT facility.Type,
```

```
      SUM(equipment.Electricity_consumption) AS
Total_Consumption
FROM facility
INNER JOIN equipment
WHERE facility.F_ID = equipment.Facility_F_ID
GROUP BY facility.Type
ORDER BY 2
```

Query 6. Calculate the total labor costs per department (COMPLEX):

```
SELECT employee.Department_D_number,
      COALESCE((SUM(`white-collar`.Salary) +
SUM(`white-collar`.Bonus)), SUM(`blue-collar`.Wage *
`blue-collar`.Hours * 52)) AS Total_labor_cost
FROM employee
LEFT OUTER JOIN `white-collar` ON employee.E_SSN =
`white-collar`.WC_E_SSN
LEFT OUTER JOIN `blue-collar` ON employee.E_SSN =
`blue-collar`.BC_E_SSN
GROUP BY employee.Department_D_number
```

Query 7. List all employees who have children and have worked overtime (>40 hours for a week) or have earned a bonus (COMPLEX):

```
SELECT employee.E_SSN AS SSN,
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`
FROM employee,
     `blue-collar`
WHERE (40 < ANY
          (SELECT `blue-collar`.hours
           FROM `blue-collar` BC
           WHERE employee.E_SSN = `blue-
collar`.BC_E_SSN))
  AND (1 <
          (SELECT COUNT(*)
           FROM dependent
           WHERE employee.E_SSN =
dependent.Employee_E_SSN))
UNION
SELECT employee.E_SSN AS SSN,
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`
FROM employee,
     `white-collar`
WHERE (0 < ANY
```

```
          (SELECT `white-collar`.bonus
           FROM `white-collar` WC
           WHERE employee.E_SSN = `white-
collar`.WC_E_SSN))
  AND (1 <
          (SELECT COUNT(*)
           FROM dependent
           WHERE employee.E_SSN =
dependent.Employee_E_SSN))
ORDER BY SSN
```

Query 8. List the departments which order an exceptional quantity of supplies (more than twice the average amount for all departments) (COMPLEX):

```
SELECT department.D_number,
       department.Name
FROM department
WHERE EXISTS
     (SELECT *
      FROM supplier
      WHERE department.D_number =
supplier.Department_D_number
      GROUP BY supplier.Department_D_number
```

```
     HAVING SUM(Price * Quantity) >
       (SELECT 2 * AVG(Price * Quantity)
        FROM supplier))
```

Query 9. List the locations which have an employee, a department, and equipment (COMPLEX):

```
SELECT department.Location,
       department.Name AS "Department",
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`,
       equipment.Type AS "Equipment"
FROM department,
     equipment,
     employee
WHERE EXISTS
    (SELECT *
     FROM facility F
     WHERE F.Location = department.Location
       AND F.F_ID = equipment.Facility_F_ID
       AND equipment.Type IN("Phone",
                             "Printer")
       AND F.Location IN
```

```
          (SELECT employee.Address
           FROM employee E2
           WHERE E2.Department_D_number =
department.D_number
             AND EXISTS
               (SELECT *
                FROM department
                WHERE department.Manager_E_SSN =
employee.E_SSN)))
GROUP BY department.Location
ORDER BY department.Name
```

Query 10. List the desserts of quantity greater than the average quantity that have expired (expiration_date < 24.11.2014) in each facility (COMPLEX):

```
SELECT dessert.Item_I_Name,
       item.Expiration_date,
       facility.F_ID
FROM dessert,
     item,
     facility
WHERE dessert.Item_I_Name = item.I_name
  AND item.I_name IN
```

```
    (SELECT Item_I_name
     FROM facility_stores_item FSI
     WHERE facility.F_ID = FSI.Facility_F_ID
       AND FSI.Quantity >
         (SELECT AVG(Quantity)
          FROM Facility_stores_item FSI2))
  AND item.Expiration_date < '20141124'
```

Query 11. List the departments with the customers they serve (COMPLEX):

```
SELECT department.Name,
       customer.Name
FROM (((department
        INNER JOIN employee ON(department.D_number =
employee.Department_D_number))
       INNER JOIN dessert ON(employee.Dessert_D_ID =
dessert.D_ID))
      INNER JOIN dessert_soldto_customer
ON(dessert.D_ID =
dessert_soldto_customer.Dessert_D_ID))
INNER JOIN customer
ON(dessert_soldto_customer.Customer_C_ID =
customer.C_ID)
WHERE customer.Address IN
```

```
    (SELECT customer.Address
     FROM customer)
ORDER BY department.Name
```

Query 12. List the items which each department has access to in its facility (COMPLEX):

```
SELECT item.I_name,
       department.Name,
       department.Location
FROM (((item
        INNER JOIN facility_stores_item ON(item.I_name
= facility_stores_item.Item_I_name))
       INNER JOIN facility
ON(facility_stores_item.Facility_F_ID = facility.F_ID))
      INNER JOIN department_housedby_facility
ON(facility.F_ID =
department_housedby_facility.Facility_F_ID)
      INNER JOIN department ON
(department_housedby_facility.Department_D_number =
department.D_number))
WHERE department.Location IN
    (SELECT facility.Location
     FROM facility)
```

```
ORDER BY department.Location
```

Optimization:

Query 3:

Place the largest table as the last one in JOIN list.

Before optimization:

```
SELECT department.Name,
       SUM(expense.Amount) AS Total_Expense
FROM ((department
       INNER JOIN department_incurs_expense)
      INNER JOIN expense)
WHERE ((department.D_number =
department_incurs_expense.Department_D_number)
       AND (department_incurs_expense.Expense_Ex_ID =
expense.Ex_ID))
GROUP BY department.Name
ORDER BY 1
```

After optimization:

```
SELECT department.Name,
       SUM(expense.Amount) AS Total_Expense
```

```
FROM ((expense
       INNER JOIN department_incurs_expense)
      INNER JOIN department)
WHERE ((expense.Ex_ID =
department_incurs_expense.Expense_Ex_ID)
       AND
(department_incurs_expense.Department_D_number =
department.D_number))
GROUP BY department.Name
ORDER BY 1
```

Query 7:

In conditions, replace ALL or ANY with an expression using MIN and/or MAX.

Before optimization:

```
SELECT employee.E_SSN AS SSN,
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`
FROM employee,
     `blue-collar`
WHERE (40 < ANY
         (SELECT `blue-collar`.hours
          FROM `blue-collar` BC
```

```
          WHERE employee.E_SSN = `blue-
collar`.BC_E_SSN))
  AND (1 <
         (SELECT COUNT(*)
          FROM dependent
          WHERE employee.E_SSN =
dependent.Employee_E_SSN))
UNION
SELECT employee.E_SSN AS SSN,
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`
FROM employee,
     `white-collar`
WHERE (0 < ANY
         (SELECT `white-collar`.bonus
          FROM `white-collar` WC
          WHERE employee.E_SSN = `white-
collar`.WC_E_SSN))
  AND (1 <
         (SELECT COUNT(*)
          FROM dependent
```

```
          WHERE employee.E_SSN =
dependent.Employee_E_SSN))
ORDER BY SSN
```

After optimization:

```
SELECT employee.E_SSN AS SSN,
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`
FROM employee,
     `blue-collar`
WHERE (40 <
         (SELECT MIN(`blue-collar`.hours)
          FROM `blue-collar` BC
          WHERE employee.E_SSN = `blue-
collar`.BC_E_SSN))
  AND (1 <
         (SELECT COUNT(*)
          FROM dependent
          WHERE employee.E_SSN =
dependent.employee_E_SSN) )
UNION
SELECT employee.E_SSN AS SSN,
```

```
       CONCAT(employee.First_name, employee.Last_name)
AS `Name`
FROM employee,
     `white-collar`
WHERE (0 <
          (SELECT MIN(`white-collar`.bonus)
           FROM `white-collar` WC
           WHERE employee.E_SSN = `white-
collar`.WC_E_SSN) )
  AND (1 <
          (SELECT COUNT(*)
           FROM dependent
           WHERE employee.E_SSN =
dependent.Employee_E_SSN) )
ORDER BY SSN
```

Query 12:

1) Place the largest table as the last one in JOIN list.

2) Filter tables with WHERE before doing JOIN.

Before optimization:

```
SELECT item.I_name,
```

```
       department.Name,
       department.Location
FROM (((item
        INNER JOIN facility_stores_item ON(item.I_name
= facility_stores_item.Item_I_name))
       INNER JOIN facility
ON(facility_stores_item.Facility_F_ID = facility.F_ID))
      INNER JOIN department_housedby_facility
ON(facility.F_ID =
department_housedby_facility.Facility_F_ID)
      INNER JOIN department ON
(department_housedby_facility.Department_D_number =
department.D_number))
WHERE department.Location IN
    (SELECT facility.Location
     FROM facility)
ORDER BY department.Location
```

After optimization:

```
SELECT item.I_name,
       filtered_department.Name,
       filtered_department.Location
```

```
FROM (facility
      INNER JOIN department_housedby_facility
ON(facility.F_ID =
department_housedby_facility.Facility_F_ID)
      INNER JOIN
        (SELECT *
         FROM department
         WHERE department.Location IN
             (SELECT facility.Location
              FROM facility)) AS filtered_department ON
(department_housedby_facility.Department_D_number =
filtered_department.D_number)
      INNER JOIN facility_stores_item ON(facility.F_ID
= facility_stores_item.Facility_F_ID)
      INNER JOIN item
ON(facility_stores_item.Item_I_Name = item.I_Name))
ORDER BY filtered_department.Location
```